

What is claimed is:

- Sub  
A2
1. A method for rendering graphics data, the method comprising:  
receiving a scene graph, wherein the scene graph comprises a hierarchical group of objects;  
5 generating a plurality of data structures, wherein each data structure corresponds to a particular type of object in the scene graph;  
creating at least one thread that operates on each data structure, wherein at least a subset of the threads are configured to generate messages to communicate state changes to one or more of the data structures; and  
10 generating multiple renderer threads that each render to a single canvas.
  2. The method of claim 1, further comprising guaranteeing that opaque objects render before transparent objects.
  - 15 3. The method of claim 1, further comprising granting each renderer thread a separate context..
  4. A method for rendering graphics data, the method comprising:  
receiving a scene graph, wherein the scene graph comprises a hierarchical group of objects;  
20 generating a plurality of data structures, wherein each data structure corresponds to a particular type of object in the scene graph;  
creating at least one thread that operates on each data structure, wherein at least a subset of the threads are configured to generate messages to communicate state changes to one or more of the data structures; and  
25 generating multiple renderer threads, wherein each renderer thread renders to a different canvas.
  5. A method for rendering graphics data, the method comprising:  
receiving a scene graph, wherein the scene graph comprises a hierarchical group of objects;  
30 generating a plurality of data structures, wherein each data structure corresponds to a particular type of object in the scene graph;

creating at least one thread that operates on each data structure, wherein at least a subset of the threads are configured to generate messages to communicate state changes to one or more of the data structures; and  
generating multiple renderer threads, wherein each renderer resides on a separate graphics device.

6. The method of claim 5, wherein each renderer thread executed in parallel.

7. The method of claim 5, wherein each renderer thread is configured to issue swap buffer commands on all canvases associated with a single view at the same time.

8. The method of claim 5, wherein each renderer thread is configured to issue swap buffer commands on all canvases associated with a single view at substantially the same time.

9. The method of claim 5, wherein each renderer thread is configured to issue swap buffer commands on all canvases associated within as close to each other as possible.

10. A computer program embodied on a computer-readable medium, wherein the computer program is configured to implement the method of claim 1.

11. The computer program of claim 10, wherein the computer program is an application program interface (API).

12. The computer program of claim 10, wherein the computer program is an application program interface (API) for the Java programming language.

13. A computer program embodied on a computer-readable medium, wherein the computer program is configured to:

receive a scene graph comprising a plurality of graphical objects;

traverse the scene graph;

generate a plurality of data structures corresponding to the scene graph, wherein at least one of the data structures is a render bin;

determine which of a plurality of graphical objects are to be rendered;  
create entries in the render bin that correspond to graphical objects that are to be rendered; and  
submit the graphical objects listed in the render bin for rendering.

14. The computer program of claim 12, further comprising rendering the graphical objects listed  
in the render bin.

15. The computer program of claim 12, further comprising generating a rendering thread,  
wherein the rendering thread is configured to render the contents of the render bin.

16. The computer program of claim 12, wherein the entries in the render bin are pointers to  
memory corresponding to the graphical objects that are to be rendered.

17. The computer program of claim 12, wherein each data structure has a message queue  
configured to receive and store messages including data for updating the data structure.

18. The computer program of claim 14, wherein the scene graph is traversed only once, and  
wherein said rendering is performed more than once.

19. The computer program of claim 12, wherein additions to the scene graph generate updates to  
one or more of the data structures.

20. The computer program of claim 12, wherein the render bin has one or more render threads  
associated with it, wherein each render thread is configured to render a portion of the render  
bin.

21. The computer program of claim 12, wherein the render bin comprises an array of pointers  
pointing to the graphical objects that are to be rendered.

22. The computer program of claim 12, wherein the render bin comprises an array of graphical  
objects that are to be rendered.

23. The computer program of claim 12, wherein the render bin comprises a list of pointers pointing to the graphical objects that are to be rendered.

24. A method for managing data that is to be rendered, the method comprising:

5 receiving a scene graph comprising a plurality of graphical objects;

generating a plurality of data structures corresponding to the scene graph, wherein at least one of the data structures is a render bin;

determining which of the graphical objects are to be rendered;

creating entries in the render bin that correspond to graphical objects that are to be rendered; and

10 submitting the graphical objects listed in the render bin for rendering.

25. The method of claim 24, wherein the data structures include: a geometry data structure, a transform data structure, and rendering environment data structure.

15 26. The method of claim 24, further comprising rendering the graphical objects having entries listed in the render bin.

20 27. The method of claim 24, further comprising generating a rendering thread, wherein the rendering thread is configured to render the graphical objects having entries in the render bin.

28. The method of claim 24, wherein the entries in the render bin are pointers to memory locations storing the graphical objects that are to be rendered.

25 29. The method of claim 24, wherein each data structure has a message queue configured to receive and store messages, wherein the messages include data for updating the data structure.

30 30. The method of claim 24, wherein the scene graph is traversed only once, and wherein said rendering is performed more than once.

095000-0110  
FBI/DOJ



39. The method of claim 37, wherein each data structure that is not a render bin has at least one working thread, wherein each working thread is configured to update the corresponding data structure.

5 40. The method of claim 35, wherein at least one of the data structures is a rendering environment data structure.

41. The method of claim 35, wherein at least one of the data structures is a transform data structure.

10 42. The method of claim 35, wherein at least one of the data structures is a geometry data structure.

15 43. The method of claim 35, wherein at least one of the data structures is a behavior structure that stores entries relating to object behavior.

05750000-01101  
TOTFO-00009260